Nagarjuna College of Engineering & Technology, Bengaluru

An Autonomous Institute, Affiliated to VTU Belagavi

2022 Batch

Scheme & Syllabus
of
IV Sem

As per the NEP 2020 Guidelines,
Choice-Based Credit System
&
Outcome-Based Education

CSE (Data Science)

Academic Year 2023-2024

Vision

To prepare the next generation practitioners and researcher for data centric world by bringing together interdisciplinary faculty across the globe.

Mission

M1: To provide Skill Based Education to master the students in problem solving and analytical skills to enhance their niche expertise in the field Data Science

M2: To educate the students with latest technologies to update their knowledge in the field of Data Science

M3: To enable students to experience the Content Based Learning with premier quality data science education, research and industrial collaboration

M4: To enable students to become leaders in the Industry and Academia Nationally as well as internationally

M5: To guide students in research on Data Science, with the aim of having an ethical impact on society by tackling societal grand challenges

PROGRAM OUTCOMES (POs): Graduates of the Computer Science and Engineering – Data Science Program will be able to achieve the following

POs:

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and Computer Science and Engineering principles to the solution of complex problems in Computer Science and Engineering.

PO2: Problem Analysis: Identify, formulate, research literature, and analyses complex Computer Science and Engineering problems reaching substantiated conclusions using first principles of mathematics and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex Computer Science and Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of Complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions related to Computer Science and Engineering problems.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex Computer Science and Engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Computer Science and Engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional Computer Science and Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the Computer Science and Engineering practice.

PO9: Individual and Team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex Computer Science and Engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage Computer Science and Engineering projects and in multidisciplinary environments.

PO12: Life Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcome (PSO)

PSO1: Ability to analyse complex computing issues and apply the principles to achieve related solution.

PSO2: Ability to design, implement and evaluate computing based solutions to meet range of computing requirements based in the data science.

PSO3: Ability to effectively communicate within diverse work group related to professional framework.

Program Educational Objectives (PEOs)

PEO 1: To make students competent for higher studies and employable, to meet industrial requirements.

PEO 2: To develop students having core competence in science, mathematics and fundamentals of Data Science to address ever changing industrial requirements globally.

PEO 3: To create academically conducive environment to learn engineering skills in the domains such as Data Analytics, Data Modelling, Data Visualization and Allied Technologies.

PEO 4: To enrich students with professional ethics, leadership qualities, and entrepreneurial skills.

PEO 5: An ability to engage in lifelong learning for effective adaptation to technological developments.

Nagarjuna College of Engineering and Technology, Bangalore

B.E. in CSE (Data Science)

Scheme of Teaching and Examinations 2022

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

(Effective from the academic year 2022-23)

IV SEMESTER

Sl. No	Course and Course Code		Course Title	Teaching Department (TD) and Question Paper Setting Board (PSB)	Teaching Hours /Week				Examination				Credits
					Theory Lectur	Tutorial	Practical	Self -Study	Duration inhours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC/ BSC	22CDT41	Software Engineering and Project Management		3	0	0		03	50	50	100	3
2	IPCC	22CDI42	Design & Analysis of Algorithms		3	0	2		03	50	50	100	4
3	IPCC	22CDI43	Data Wrangling using Python		3	0	2		03	50	50	100	4
4	PCCL	22CDL44	Web Technology based Mini project		0	0	2		03	50	50	100	1
5	ESC	22CDT45x	ESC/ETC/PLC		3	0	0		03	50	50	100	3
6	AEC/ SEC	22CDL46x	Ability Enhancement Course/Skill Enhancement Course- IV	TD and PSB: Concerned department	If the course is Theory: 1	0	0		01	50	50	100	1
					If the course is a lab: 0	0	2		02				
7	BSC	22BET47	Biology For Engineers	TD / PSB: Any Branch/ BT, CHE,	3	0	0		03	50	50	100	3
8	UHV	22UHV48	Universal human values course	Any Department	1	0	0		01	50	50	100	1
9		22NS49	National Service Scheme (NSS)	NSS coordinator	0	0	2			100	---	100	0
		22PE49	Physical Education (PE) (Sports and Athletics)	Physical Education Director									
		22YO49	Yoga	Yoga Teacher									
									Total	**500**	**400**	**900**	**20**

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **MC**: Mandatory Course (Non-credit), **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. K : This letter in the course code indicates common to all the stream of engineering.

Ability Enhancement Course / Skill Enhancement Course - IV			
22CDL46A	Git and GitHub	22CDL46B	Object Oriented Modelling lab using Star UIML
Engineering Science Course (ESC/ETC/PLC)			
22CDT45A	Discrete Mathematics and Graph Theory	22CDT45C	Introduction to Computer Graphics and Visualization
22CDT45B	Agile Methodologies	22CDT45D	Unix System Programming

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching– Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governingthe Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall

not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the courses is mandatory for the award of degree.

SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Course Code	**22CDT41**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Required Knowledge of: Basics of any programming language, software types, functions, and steps of software development

Course Learning Objectives

CLO1. Contrast use of Software Engineering and associated processes using standard models.
CLO2. Identify the software functions and associated component to design architectural framework.
CLO3. Decide the separation of concern and design relevant processes for the required operations.
CLO4. Prepare test cards to measure project performance accomplishing specified requirements

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module – I

Introduction:
Professional software development, Software engineering ethics, Case studies.
Software Processes: Software Process models: The Waterfall model – A Case study, Incremental development, Reuse-oriented software engineering, Process activities: Software specification, Software design and implementation, Software validation, Coping with Change: Prototyping, Incremental Delivery, Boehm's Spiral Model.

08 Hours

Module – II

Requirements Engineering: Functional and non-functional requirements: Functional requirements. Non-functional requirements, Introduction to Requirements specification.
Agile Software Development: Agile methods- Plan driven and Agile Development, Introduction to Extreme Programming. **08 Hours**

Module – III

Design and Implementation: Object-oriented design using UML: System Context and Interaction, Architectural design, Object Class identification, design Models, Interface Specification, Design Patterns, Implementation issues, Open Source development. **08 Hours**

Module – IV

Software Testing: Development Testing: Unit Testing, Choosing Unit Test Cases, Component Testing, System Testing, Test Driven Development, Release Testing: Requirements Based Testing, Scenario Testing, Performance Testing, User Testing. A Demo of Selenium**.** **08 Hours**

Module - V

Introduction to Project Management:
Introduction, Project and Importance of Project Management, Contract Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some ways of categorizingSoftwareProjects,Stakeholders,SettingObjectives,BusinessCase,ProjectSuccessandFailure,ManagementandManagementControl,ProjectManagementlifecycle,
Textbook2:Chapter1:1.1to1.17

08 Hours

Teaching-Learning Process	Chalk and board, Active Learning, Demonstration
Sl. No.	**Self-Study Topics**
1	Identification of requirements for any common software in use by business domain and the advantages.
2	Classification of functional and non-functional requirements of any software used in business domain. Software Architectural patterns, implementation and uses.
3	Object oriented software and UML: Business use case Design and Activity diagrams
4	Software testing ISO 9001 series – Guidelines applicable to software industry
5	Software Quality & Performance: Git-Hub based topics with ref. link: https://github.com/ICTU/quality-time

Course Outcomes
At the end of the course, the student will be able to:

CO1. Apply the professional practice for software development requirements for associated processes, and feasibility and decide the suitable model of software.

CO2. Analyze the software design accumulating information and the functional components for the development.

CO3. Evaluate the software testing methods. to check the accuracy based on the analysis of contextual requirements.

CO4. Demonstrate software that matches industry needs and adapts the changes based on demand for the continuous quality improvement.

CO5. Design a course project by applying the learning's inculcated throughout the course.

Assessment Details (both CIE and SEE)

Evaluation Type	**Component**	**Max. Marks**	**Marks Reduced To**	**Min Marks**	**Evaluation Details**
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled-down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

Text books

1. Ian Summerville: Software Engineering, Pearson Education, 9th Edition onwards
2. Bob Hughes, Mike Cotterell, Rajib Mall: Software Project Management, 6th Edition, Mc Graw Hill Education, 2018.

Reference:

1. Roger .S. Pressman: Software Engineering-A Practitioners approach, 8th Edition and above, Tata McGraw Hill
2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India, 2009 onwards
3. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML, 2nd Edition, Pearson Education,2005.
4. Deepak Gaikwad, Viral Thakkar, DevOps Tools From Practitioner‘s View point, Wiley.

Web links and Video Lectures (e-Resources):

1. NPTEL: https://nptel.ac.in/courses/106105182
2. SWAYAM: https://onlinecourses.swayam2.ac.in/cec20_cs07/preview
3. IIT Chennai: https://onlinedegree.iitm.ac.in/course_pages/BSCCS3001.html
4. https://www.youtube.com/watch?v=WxkP5KR_Emk&list=PLrjkTql3jnm9b5nr-ggx7Pt1G4UAHeFlJ
5. http://elearning.vtu.ac.in/econtent/CSE.php
6. http://elearning.vtu.ac.in/econtent/courses/video/CSE/15CS42.html
7. https://nptel.ac.in/courses/128/106/128106012/(DevOps)

DESIGN AND ANALYSIS OF ALGORITHMS

Course Code	**22CDI42**	**CIE Marks**	**50**
Teaching Hours/Week (L: T: P: S) (3:0:2:0)	**3:0:2:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40 T + 20 P**	**Total Marks**	**100**
Credits	**04**	**Exam Hours**	**03**

Prerequisites: Data Structures, Algorithms, Algebra.

Course Learning Objectives:

CLO1. Explain the methods of analysing the algorithms and to analyze performance ofalgorithms.
CLO2. State algorithm‘s efficiencies using asymptotic notations.
CLO3. Solve problems using algorithm design methods such as the brute force method, greedy method, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking and branch and bound.
CLO 4. Choose the appropriate data structure and algorithm design method for a specified application.
CLO 5. Introduce P and NP classes.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) does not mean only traditional lecture method, but different type of teaching methods may be adopted to develop the outcomes.
2. Show Video/animation films to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students‘ Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
6. Topics will be introduced in a multiple representation.
7. Show the different ways to solve the same problem and encourage the students to come up withtheir own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module – I

Introduction: What is an Algorithm?, Fundamentals of Algorithm problem solving, Fundamentals of analysis of algorithm efficiency, Asymptotic notations and basic efficiency classes, Mathematical analysis of Recursive and Non Recursive Algorithms.
Brute Force Approaches: Introduction, Selection Sort, Bubble Sort, Sequential Search, Brute force String Matching.

08 Hours

Module – II

Divide and Conquer: General method, Recurrence equation for divide and conquer, solving it using Master‘s theorem. , Divide and Conquer algorithms and complexity Analysis - Finding the maximum & minimum, Binary search, Merge sort, Quick sort.
Greedy Method-The General Method, Minimum-cost Spanning Tree, Prims Algorithm, Kruskal’s Algorithm, Single Source Shortest Path.

08 Hours

Module – III

Decrease and Conquer Approach: Introduction, Insertion sort, Depth First Search, Breadth First Search, Topological Sorting.
Transform and Conquer Approach: Introduction,2-3 Trees, Heaps and Heap Sort. **08 Hours**

Module – IV
Dynamic Programming: General method, Warshall's Algorithm, Floyd's Algorithm for all pair shortest path, Travelling Salesman Problem, Computing Binomial Coefficient. **Space-Time Tradeoffs**: Introduction, Sorting by Counting, Input Enhancement in String Matching-Harspool's algorithm. **08 Hours**
Module – V
Backtracking: N-Queens problem, Sum ofsubsets problem. **Branch and Bound**: Assignment Problem, Knapsack problem **Hashing**-Open Hashing, Closed Hashing **08 Hours**

Teaching-Learning Process for all modules	1. Chalk & board, Active Learning, MOOC, Problem based Learning. 2. Laboratory Demonstration.

PRACTICAL COMPONENTS

Sl. No	Experiments
1	Write a C++ program to sort the elements by using quick sort.
2	Write a C++ program to sort the elements using Merge Sort
3	From a given vertex in a weighted connected graph, Find the shortest path to other vertices using Dijkstra's algorithm.
4	Implement 0/1 Knapsack problem using Dynamic Programming.
5	Find the Minimum cost Spanning Tree of a given undirected graph using Prim's Algorithm.
6	Compute the transitive closure of a given directed graph using Warshall's Algorithm,
7	Implement all pair shortest path problems using Floyd's algorithm.
8	Print all nodes reachable from a given starting node in a digraph using the BFS Method.
9	Check whether a given graph is connected or not using the DFS Method.
10	Implement N Queen's problem using Backtracking.

Course Outcomes

At the end of the course, the student will be able to:

CO1: Apply and identify asymptotic notations and basic efficiency classes.
CO2: Solve problems using various techniques like greed and divide and conquer.
CO3: Compute problems using various techniques like decrease-and-conquer and transfer-and-conquer.
CO4: Interpret different algorithms like TSP, Floyd's, etc. to solve real-world problems.
CO5: Design and develop solutions for n-Queens problem, Subset-sum problem, Assignment problem,
Knapsack problem etc.

Assessment Details (both CIE and SEE)

IPCC / Integrated Courses : 4 Credits and 3 Credits Courses						
Evaluation Type		Component	Max Marks	Marks reduced to	Min. Marks	Evaluation Details
Theory Component	Internal Assessment Tests(IAT)	IAT-1	25	15	10	Average of two IATs, Scaled down to 15 marks
		IAT-2	25			
	Comprehensive Continuous Evaluations (CCE)	CCE-1	10	10		Minimum of two assessment methods as per 22OB4.2 of regulations. Average of CCEs, scaled down to 10 marks.
		CCE-2	10			
	Total CIE - Theory			**25**	**10**	Scale down marks of IAT and CCE to 25
Laboratory Component	Practicals and Lab Records	-	15	25	10	Conduction of experiments and preparation of Lab records, etc
	Lab Test	50	10			One test to be conducted after the completion of all lab experiments.
	Total CIE – Practicals			**25**	**10**	
Total CIE (Theory + Lab)				**50**	**20**	
SEE			100	**50**	**18**	Conducted for 100 marks and scaled down to 50.
CIE + SEE				**100**	**40**	
The minimum marks to be secured in CIE to appear for SEE shall be 10(40% of maximum marks: 25) in theory component and 10(40% of maximum marks: 25) in the practical component. The laboratory component of the IPCC/Integrated course shall be for CIE only. However, in SEE, the questions from the laboratory component shall be included in the question paper.						

Suggested Learning Resources:

Textbooks

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson.
2. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.

Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

Web links and Video Lectures (e-Resources):

1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS43.html
2. https://nptel.ac.in/courses/106/101/106101060/
3. http://elearning.vtu.ac.in/econtent/courses/video/FEP/ADA.html
4. http://cse01-iiith.vlabs.ac.in/
5. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course= Intro To Algorithms

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification, Peasant, wolf, goat, cabbage puzzle, Konigsberg bridge puzzle etc
2. Demonstration of solution to a problem through programming.
3. Leetcode and HackerRank Problems,

DATA WRANGLING USING PYTHON

Course Code	**22CDI43**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:2:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40 T + 20 P**	**Total Marks**	**100**
Credits	**04**	**Exam Hours**	**03**

Prerequisites: Python Programming.

Course Learning Objectives

- To learn different methods for Data Cleanup
- To learn basics of Python.
- To learn Data aggregation Techniques.
- To learn about packages Numpy and Pandas
- To learn concepts of Web scraping.

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students‘ Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module - I

INTRODUCTION TO DATA WRANGLING: What Is Data Wrangling?- Importance of Data Wrangling -How is Data Wrangling performed?- Tasks of Data Wrangling-Data Wrangling Tools-Introduction to Python-Python Basics-Data Meant to Be Read by Machines-CSV Data-JSON Data-XML Data.

08 Hours

Module - II

WORKING WITH EXCEL FILES AND PDFS: Installing Python Packages-Parsing Excel Files-Parsing Excel Files -Getting Started with Parsing-PDFs and Problem Solving in Python-Programmatic Approaches to PDF Parsing-Converting PDF to Text-Parsing PDFs Using pdf miner-Acquiring and Storing Data-Databases: A Brief Introduction-Relational Databases: MySQL and Postgre SQL-Non-Relational Databases: No SQL-When to Use a Simple File-Alternative Data Storage.

08 Hours

Module - III

DATA CLEANUP: Why Clean Data?- Data Cleanup Basics-Identifying Values for Data Cleanup-Formatting Data-Finding Outliers and Bad Data-Finding Duplicates-Fuzzy Matching-RegEx Matching-Normalizing and Standardizing the Data-Saving the Data-Determining suitable Data Cleanup-Scripting the Cleanup- Testing with New Data. **08 Hours**

Module - IV
DATA EXPLORATION AND ANALYSIS: Exploring Data-Importing Data-Exploring Table Functions-Joining Numerous Datasets-Identifying Correlations-Identifying Outliers-Creating Groupings-Analyzing Data-Separating and Focusing the Data- Presenting Data-Visualizing the Data-Charts-Time-Related Data-Maps-Interactive-Words-Images, Video, and Illustrations-Presentation Tools-Publishing the Data-Open Source Platforms. **08 Hours**
Module - V
WEB SCRAPING: What to Scrape and How-Analyzing a Web Page-Network/Timeline-Interacting with JavaScript-In-Depth Analysis of a Page-Getting Pages-Reading a Web Page-Reading a Web Page with LXML-XPath-Advanced Web Scraping-Browser-Based Parsing-Screen Reading with Selenium-Screen Reading with Ghost.Py- Spidering the Web-Building a Spider with Scrapy-Crawling Whole Websites with Scrapy. **08 Hours**

Teaching-Learning Process for all modules	Chalk and board, Active Learning, PPT Based presentation, Video

PRACTICAL COMPONENTS

Sl. No	Experiments
1	Write a Python script to read each row from a given csv file and print a list of strings.
2	Write a Python program to read a given CSV file as a dictionary.
3	Write a Python program to convert Python dictionary object (sort by key) to JSON data. Print the object members with indent level 4
4	Write the python script to Read the XML file
5	Write a Pandas program to import excel data into a Pandas data frame and process the following a. Get the data types of the given excel data b. Display the last ten rows. c. Insert a column in the sixth position of the said excel sheet and fill it with NaN values
6	Develop the python script to parse the pdf files using pdfminer.
7	Write a Python data wrangling scripts to insert the data into SQLite database
8	Develop the Python Shell Script to do the basic data cleanup on excel data to a. Check duplicates and missing data b. Eliminate Mismatches c. Cleans line breaks, spaces, and special characters
9	Develop a python program to parse data representing information about individuals. df1 contains the columns 'ID' and 'Name', while df2 contains the columns 'ID' and 'Age'. Perform the following joins and describe the resulting DataFrames: a. inner join b. left join c. right join d. outer join
10.	Develop a python program to use the merge operation in order to merge the data and provide it meaning for the following scenario A teacher has two types of Data, the first type of Data consists of Details of Students and the Second type of Data Consist of Pending Fees Status which is taken from the Account Office.

11	Write a python script for the following scenario There is a Car Selling company and this company have different Brands of various Car Manufacturing Company like Maruti, Toyota, Mahindra, Ford, etc., and have data on where different cars are sold in different years. So the Company wants to wrangle only that data where cars are sold during the year 2010.(Grouping)

Course Outcomes:

On completion of this course, the students will be able to,

- **CO 1**: Identify and execute the basic data format.
- **CO 2**: Analyze the computations with Excel and pdf files.
- **CO 3**: Apply the concepts of data cleanup.
- **CO 4**: Analyze and explore the Image and video data.
- **CO 5**: Apply the concepts of web scraping.

Assessment Details (both CIE and SEE)

IPCC / Integrated Courses : 4 Credits and 3 Credits Courses

Evaluation Type		Component	Max Marks	Marks reduced to	Min. Marks	Evaluation Details
Theory Component	Internal Assessment Tests(IAT)	IAT-1	25	15	10	Average of two IATs, Scaled down to 15 marks
		IAT-2	25			
	Comprehensive Continuous Evaluations (CCE)	CCE-1	10	10		Minimum of two assessment methods as per 22OB4.2 of regulations. Average of CCEs, scaled down to 10 marks.
		CCE-2	10			
	Total CIE - Theory			**25**	**10**	Scale down marks of IAT and CCE to 25
Laboratory Component	Practicals and Lab Records	-	15	25	10	Conduction of experiments and preparation of Lab records, etc
	Lab Test	50	10			One test to be conducted after the completion of all lab experiments.
	Total CIE – Practicals			**25**	**10**	
Total CIE (Theory + Lab)				**50**	**20**	
SEE			100	**50**	**18**	Conducted for 100 marks and scaled down to 50.
CIE + SEE				**100**	**40**	

The minimum marks to be secured in CIE to appear for SEE shall be 10(40% of maximum marks: 25) in theory component and 10(40% of maximum marks: 25) in the practical component. The laboratory component of the IPCC/Integrated course shall be for CIE only. However, in SEE, the questions from the laboratory component shall be included in the question paper.

Text Books:

1. Jacqueline Kazil & Katharine Jarmul,‖ Data Wrangling with Python‖, O'Reilly Media, Inc,2016.

References:

1. Dr. Tirthajyoti Sarkar, Shubhadeep,‖ Data Wrangling with Python: Creating actionable data from raw sources‖, Packt Publishing Ltd,2019.
2. Stefanie Molin,‖ Hands-On Data Analysis with Pandas‖, Packt Publishing Ltd,2019.
3. Allan Visochek,‖ Practical Data Wrangling‖, Packt Publishing Ltd,2017.
4. Tye Rattenbury, Joseph M. Hellerstein, Jeffrey Heer, Sean Kandel, Connor Carreras,‖ Principlesof Data Wrangling: Practical Techniques for Data Preparation‖, O'Reilly Media, Inc, 2017.

E-Books:

1. http://www.gbv.de/dms/ilmenau/toc/827365454.PDF

E-Resources:

1. **https://www.kaggle.com/datasets**

MOOC:

1. https://www.udemy.com/course/data-wrangling-with-python/
2. http://www.openculture.com/free-online-data-science-courses
3. https://www.classcentral.com/course/dataanalysiswithpython-11177

DISCRETE MATHEMATICS AND GRAPH THEORY

Course Code	**22CDT45A**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**2:2:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Pre-requisite: Basic mathematical concepts, including arithmetic, algebra, geometry, and calculus.

Course Objectives

The goal of the course DISCRETE MATHEMATICS AND GRAPH THEORY - 22CDT45A is to

- Understand an intense foundational introduction to fundamental concepts in discrete mathematics.
- Interpret, identify, and solve the language associated with logical structure, sets, relations and functions.
- Have insight into graph theoretical concepts and graph algorithms.
- Apply the knowledge of modular arithmetic to computer algorithms

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the delivered lessons shall develop students' theoretical and applied mathematical skills.
2. State the need for Mathematics with Engineering Studies and Provide real-life examples.
3. Support and guide the students for self–study.
4. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress.
5. Encourage the students for group learning to improve their creative and analytical skills.
6. Show short related video lectures in the following ways:
   - As an introduction to new topics (pre-lecture activity).
   - As a revision of topics (post-lecture activity).
   - As additional examples (post-lecture activity).
   - As an additional material of challenging topics (pre-and post-lecture activity).
   - As a model solution of some exercises (post-lecture activity).

Module - I

Sub title: Fundamentals of Logic.

Basic connectives and truth tables, Logical equivalence – The laws of Logic, Logical implication – Rules of Inference. Fundamentals of Logic contd.: The Use of Quantifiers, Quantifiers, Definitions, and the Proofs of Theorems. **(8 Hours)**

Self-study: Problems on Logical equivalence.

(RBT Levels: L1, L2 and L3)

Module - II

Sub title: Relations, and Functions.

Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. Function Composition, and Inverse Functions.

Relations: Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **(8 Hours)**

Self-study: The Pigeon-hole Principle, problems and its applications.

(RBT Levels: L1, L2 and L3)

Module - III

Sub title: Introduction to Graph Theory

Introduction to Graph Theory: Introduction- Basic definition – Application of graphs-finite, infinite and bipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. Paths and circuits – Isomorphism, sub graphs, walks, paths and circuits, connected graphs, directed graphs, disconnected graphs and components, Eulerian and Hamiltonian graphs, Travelling salesman problem.

Self-study: Fleury's algorithm **(8 Hours)**
(RBT Levels: L1, L2 and L3)

Module - IV

Sub title: Trees and Graph Algorithms

Trees – properties, pendant vertex, Distance and centres in a tree - Rooted and binary trees, counting trees, spanning trees, Prim's algorithm and Kruskal's algorithm, Dijkstra's shortest path algorithm, Floyd-Warshall shortest path algorithm. **(8 Hours)**

Self-Study: Bellman-Ford algorithm
(RBT Levels: L1, L2 and L3).

Module - V

Sub title: Modular Arithmetic

Introduction to congruences, Linear Congruences, The Remainder theorem, Solving Polynomials, Linear Diophantine equation, System of Linear Congruences, Euler's Theorem, Wilson theorem, and Fermat's little theorem. Applications of congruences-RSA algorithm. **(8 Hours)**

Self-Study: Divisibility, GCD, Properties of Prime number, Fundamental theorem of Arithmetic.
(RBT Levels: L1, L2 and L3).

Teaching-Learning Process for all modules	Chalk and talk method / Power Point Presentation

Course outcomes

After successfully completing the course, the students will be able to :

- Apply the concepts of logic for effective computation and relating problems in the Engineering domain.
- Analyze the concepts of functions and relations to various fields of Engineering.
- Analyze the concepts of Graph Theory and its algorithms for various applications of Computational sciences.
- Apply modular arithmetic to computer algorithms

Assessment Details (both CIE and SEE)

Evaluation Type		Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Theory Component	**Internal Assessment Test (IAT)**	IAT-1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
		IAT-2	25			
	Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Any two Assessment methods as per 22OB4.2 of regulations. Average of two CCEs, scaled down to 25 marks
		CCE-2	25			
Total CIE			**-**	**50**	**20**	
SEE			**100**	**50**	**18**	Conducted for 100 marks and scaled down to 50.
CIE + SEE			**-**	**100**	**40**	

Evaluation Details:

Semester End Examination

Theory SEE will be conducted as per the scheduled time table, with common question papers for the subject (**duration 03 hours).**

The question paper will have ten questions. Each question is set for 20 marks.

There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub- questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. **Ralph P. Grimaldi and B V Ramana**: Discrete and Combinatorial Mathematics- An Applied Introduction, Pearson Education, Asia, Fifth edition – 2007. ISBN 978-81-7758-424-0.
2. **Narsingh Deo**, Graph theory, PHI,1979.
3. **David M Burton:** "Elementary Number Theory" Mc Graw Hill, 7th Ed., 2017.

Reference Books:

1. **Kenneth H. Rosen**: Discrete Mathematics and its Applications, Tata – McGraw Hill,Sixth Edition, Sixth reprint 2008. ISBN-(13):978-0-07-064824-1.
2. **C. L. Liu and D P Mohapatra:** Elementary Discrete Mathematics, Tata- McGraw Hill,Sixth Edition, ISBN:10:0-07-066913-9.
3. **J.P. Tremblay and R. Manohar:** Discrete Mathematical Structures with Applicationsto Computer Science, Tata – McGraw Hill, 35TH reprint 2008. ISBN 13:978-0-07- 463113-3.
4. **William Stallings:** Cryptography and Network Security, Pearson Prentice Hall, 6th Ed., 2013.
5. **Ajay Kumar Chaudhuri:** Introduction to Number Theory, NCBA Publications, 2nd Ed., 2009.
6. **Thomas Koshy:** Elementary Number Theory with Applications, Harcourt Academic Press, 2nd Ed., 2008.

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

- Quizzes
- Assignments
- Seminars

AGILE METHODOLOGIES			
Course Code	**22CDT45B**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Pre-requisite: Knowledge of software engineering fundamentals.

Course Objectives:

This course will enable students to,

- Identify the limitations of traditional (waterfall) software development methodologies. Interpret how to prioritize user stories while estimating.
- Explain the concept and methodology behind Planning Poker as a technique used in Agile for estimating effort or complexity in user stories.
- Explain the core principles and values underpinning Agile methodologies, emphasizing flexibility, collaboration, iterative development, and responding to change.
- Explain the principles, roles, ceremonies, and artifacts of the Scrum framework in Agile software development methodologies.
- Explain the Agile software development lifecycle and its iterative nature, emphasizing the integration of testing throughout the development process.

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic andencourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module - I

Introduction: Need of Agile software development, agile context– Manifesto, Principles, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility.
Project Planning: Recognizing the structure of an agile team– Programmers, Managers, Customers.

08 Hours

Module - II

Project planning: User stories– Definition, Characteristics and content.

Estimation– Planning poker, Prioritizing, and selecting user stories with the customer, projecting team velocity for releases and iterations. **08 Hours**

Module - III

Project Design: Fundamentals, Design principles–Single responsibility, Open-closed, Liskov

substitution, Dependency-inversion, Interface-segregation.

Extreme Programming- Core principles, values and practices. Kanban, Feature-driven development

08 Hours

Module - IV

Design Methodologies: Need of scrum, Scrum practices –Working of scrum, Project velocity, Burn down chart, Sprint backlog, Sprint planning and retrospective, Daily scrum, Scrum roles– Product Owner, Scrum Master, Scrum Team. **08 Hours**

Module - V

Testing: The Agile lifecycle and its impact on testing, Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.

08 Hours

Teaching-Learning Process for all modules	Chalk and board, Active Learning, PPT Based presentation, Video

Course Outcomes:

On completion of this course, the students will be able to,

- Apply the Limitations of Traditional Methodologies. Prioritization of User Stories and Estimation.
- Analyze the Concept of Planning Poker. Application and Implementation of Planning Poker.
- Apply Agile Principles, Values. Emphasizing Flexibility in Agile Methodologies
- Analyze roles and responsibilities of Scrum Principles.
- Interpret the Iterative and Incremental Development, Integration of Testing Across Development Phases

Assessment Details (both CIE and SEE)

Evaluation Type	Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

Text Books:

1. Ken Schawber, Mike Beedle, "Agile Software Development with Scrum", International Edition, Pearson
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Pedro M. Santos, Marco Consolaro, and Alessandro Di Gioia, "Agile Technical Practices Distilled: A learning journey in technical practices and principles of software design", First edition, Packt Publisher.

REFERENCES

4. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams", International edition, Addison Wesley
5. Alistair Cockburn, "Agile Software Development: The Cooperative Game", 2nd Edition, Addison-Wesley

Online Courses and Video lectures

1. “Agile Software Development”, https://www.edx.org/course/agile-software-development Accessed on August 27, 2021.
2. “Agile Software Development”, https://www.coursera.org/learn/agile-software-development Accessed on August 27, 2021

INTRODUCTION TO COMPUTER GRAPHICS AND VISUALIZATION

Course Code	**22CDT45C**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Prerequisite: Knowledge in Basic mathematical preparations in linear algebra (matrix operations), geometry, trigonometry, and calculus and basic programming skills

Course Objectives:

This course will enable students to,

CLO1 : Explain hardware, software and OpenGL Graphics Primitives.
CLO2 : Illustrate interactive computer graphics using OpenGL.
CLO3 : Design and implementation of algorithms for 2D graphics Primitives and attributes.
CLO4 : Demonstrate Geometric transformations, viewing on both 2D and 3D objects.
CLO5 : Infer the representation of curves, surfaces, Color, and Illumination models.

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic andencourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module - I

Introduction to Raster Scan displays, Pixels, Frame buffer, Vector & Character generation, Random Scan systems, Display devices, Scan Conversion techniques, Line Drawing algorithms: simple DDA, Bresenham's Algorithm, Circle Drawing Algorithms: Midpoint Circle drawing and Bresenham's Algorithm, Polygon fill algorithm: Boundary-fill and Flood-fill algorithms.

08 Hours

Module - II

2-D Transformation: Translation, Rotation, Scaling, Shearing, Reflection. Inverse Transformation, Homogeneous coordinate system, Matrices Transformation, Composite Transformation. Windowing & Clipping: World Coordinate System, Screen Coordinate System, Viewing Transformation, Line Clipping & Polygon Clipping Algorithms.

08 Hours

Module - III

3-D Transformations: Translation, Rotation and Scaling. Parallel & Perspective Projection: Types of Parallel & Perspective Projection, Hidden Surface elimination: Depth comparison, Back face detection algorithm, Painter's Algorithm, Z-Buffer Algorithm. Curve generation, Bezier and B-spline methods. Basic Illumination Model: Diffuse reflection, Specular reflection, Phong Shading, Gouraud shading, Ray Tracing, Color models like RGB, YIQ, CMY, HSV. **08 Hours**

Module - IV

Visualization: Visualization of 2D/3D scalar fields: color mapping, ISO surfaces. Direct volume data rendering: ray-casting, transfer functions, segmentation. Visualization of Vector fields and flow data, Time-varying data, High-dimensional data: dimension reduction, parallel coordinates, Non-spatial data: multi-variate, tree/graph structured, text Perceptual and cognitive foundations, Evaluation of visualization methods, Applications of visualization, Basic Animation Techniques like traditional, key framing.

08 Hours

Module - V

Multimedia :Basic of multimedia, application of Multimedia, Text-Types, Unicode Standard ,text Compression, Text file formats, Audio Components, Digital Audio, Digital Audio processing, Sound cards, Audio file formats ,Audio Processing software ,Video-Video color spaces, Digital Video, Digital Video processing, Video file formats. Animation: Uses of Animation, Principles of Animation, Computer based animation, 3D Animation, Animation file formats, Animation software, Special Effects in animation, Storyboarding for Animation, Compression: Lossless/Lossy Compression techniques, Image, Audio & Video Compression, MPEG Standards ,Multimedia Architecture, Multimedia databases.

08 Hours

Teaching-Learning Process for all modules	Chalk and board, Active Learning, PPT Based presentation, Video

Course Outcomes:

On completion of this course, the students will be able to,

- Design and implement algorithms for 2D graphics primitives and attributes.
- Interpret Geometric transformations on both 2D and 3D objects.
- Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
- Design a suitable hardware and software for developing graphics packages using OpenGL.
- Apply fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering.

Assessment Details (both CIE and SEE)

Evaluation Type	Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

Text Books:

1. Donald Hearn and M.P. Becker “Computer Graphics” Pearson Pub.
2. Foley, Van Dam, Feiner, Hughes, “Computer Graphics: Principles and Practice” AddisonWesley
3. Rogers, "Procedural Elements of Computer Graphics", Tata McGraw Hill
4. Parekh “Principles of Multimedia” Tata McGraw Hill
5. Maurya, “Computer Graphics with Virtual Reality System” , Wiley India

Reference Books:

1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
2. Xiang, Plastock : Computer Graphics , sham‟s outline series, 2nd edition, TMG.
3. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning
4. M M Raikar & Shreedhara K S Computer Graphics using OpenGL, Cengage publication

List of Open Source Software/learning website:

1. https://www.opengl.org/
2. https://learnopengl.com/Getting-started/OpenGL
3. https://developer.nvidia.com/opengl

UNIX SYSTEM PROGRAMMING

Course Code	**22CDT45D**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Pre-requisite: C/C++ Programming with Fundamental OS

Course Objectives:

This course will enable students to,

1. Gain a comprehensive understanding of the UNIX operating system architecture, its components, and functionalities.
2. Develop proficiency in utilizing system calls, APIs, and libraries specific to UNIX programming.
3. Learn advanced file handling and manipulation techniques within the UNIX environment.
4. Master multi-threading and concurrent programming principles in UNIX systems.
5. Explore networking concepts and effectively implement network applications using UNIX sockets and protocols.

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, andanalyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic andencourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module - I

Introduction: UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

08 Hours

Module - II

UNIX Files and APIs: File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links. UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs.

08 Hours

Module - III

UNIX Processes and Process Control: The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. Process Control: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs,Orphaned Process Groups.

08 Hours

Module - IV

Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules,Error Logging, Client-Server Model

08 Hours

Module - V

nterprocess Communication : Overview of IPC Methods, Pipes, popen, pclose, Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores. Shared Memory, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.

08 Hours

Teaching-Learning Process for all modules	Chalk and board, Active Learning, PPT Based presentation, Video

Course Outcomes:

On completion of this course, the students will be able to,

Corse outcomes On completion of this course, the students will be able to

1. Demonstrate a comprehensive understanding of UNIX operating system architecture and its core principles.
2. Design anddDevelop proficient skills in programming using UNIX system calls and APIs.
3. Design and implement multi-threaded and concurrent applications within the UNIX environment.
4. Analyze and troubleshoot system-level problems and debugging techniques in UNIX-based applications.
5. Apply advanced file I/O operations and manipulation techniques in UNIX.

Assessment Details (both CIE and SEE)

Evaluation Type	**Component**	**Max. Marks**	**Marks Reduced To**	**Min Marks**	**Evaluation Details**
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

Text Books:

1. Unix System Programming Using C++ - Terrence Chan, PHI, 1999.
2. Advanced Programming in the UNIX Environment - Richard Stevens, Stephen A. Rago, 3nd Edition, Pearson Education / PHI, 2005.

REFERENCES

1. Advanced Unix Programming- Marc J. Rochkind, 2nd Edition, Pearson Education,2005.
2. The Design of the UNIX Operating System - Maurice.J.Bach, Pearson Education /PHI, 1987.
3. Unix Internals - Uresh Vahalia, Pearson Education, 2001.

Online Courses and Video lectures

1. https://www.cs.purdue.edu/homes/grr/SystemsProgrammingBook/Book/Chapter4-IntroductiontoUNIXSystemsProgramming.pdf
2. https://elearningatria.files.wordpress.com/2013/10/cse-vi-unix-system-programming-10cs62-notes.pdf

GIT AND GITHUB

Course Code	**22CDL46A**	**CIE Marks**	**50**
Teaching Hours /Week(L:T:P)	**(0:0:2)**	**SEE Marks**	**50**
Total Hours of Pedagogy	**12 Lab slots**	**Total Marks**	**100**
Credits	**01**	**Exam Hours**	**03**

Pre-requisite: Basic Command-Line Skills, Basic Text Editing Skills

Course objectives:

This course will enable students to:

1. Understand the concept of version control to track changes in your codebase.
2. Familiarize yourself with basic Git commands (init, add, commit, status, log).
3. Learn to create, switch, and merge branches.
4. Understand remote repositories and how to collaborate with others.
5. Learn how to handle merge conflicts when multiple contributors modify the same file.

LIST OF LABORATORY PROGRAMS

1.	Introduction to Git, Git setup: Install Git and SourceTree (on Windows only) on user machine, configure and create a workspace
2.	Creating a project: Create a account in Github and create a new repository in github. Link to Course Version Control with Git (Coursera):
3.	First commit, Branching and merge requests: Edit a file, commit and push (default IDE will be Source Tree), Create a branch, make changes, push and create merge request Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
4.	Git commands: Git config, git init, git clone, git add, git commit, git status Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
5.	Git commands: Git branch, git merge, git push, git pull, git check out Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
6.	First commit, Branching and merge requests: Edit a file, commit and push (default IDE will be Source Tree), Create a branch, make changes, push and create merge request Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
7.	Merging branches by merge: Create a branch, make conflicting changes on master and branch and merge from branch to master Link to Course Introduction to Git and GitHub (edX):

8.	Cherry-picking, Reset and revert: Use GitLab GUI to make two changes to a file on a branch and then cherry-pick it the first change to master, Change a file, commit and reset the commit. Change a file, commit and revert the commit. Link to Course Interactive Learning Platforms: Git and GitHub (Codec ademy):
9.	Issue Tracking Create an issue in an github. Link to Course Interactive Learning Platforms: Git and GitHub (Codecademy
10.	Intro to Wikis; Create a wiki page for an simple project on github. Link to Course Interactive Learning Platforms: Git and GitHub (Codecademy

Course Outcomes:

At the end of the course, the student will be able to :

1. Create the ability to track changes, revert to previous states, and manage different versions of the codebase.
2. Analyze the seamless collaboration among team members, with the ability to work on different features or fixes simultaneously through branching.
3. Demonstrate code integrity by preventing accidental overwrites and conflicts through branching and merging strategies.
4. Create clear and traceable history of changes, making it easier to identify when and why specific modifications were made.
5. Solve parallel development of features or bug fixes without disrupting the main codebase.

TEXT BOOKS:

1. "Git and GitHub Guide: The Basics" Jerry N P, Kindle Edition.
2. "Introduction to Git and GitHub" Bobby Iliev.
3. "GitHub For Dummies" Sarah Guthals, Phil Haack, and Zachary Voase, 2nd Edition, Dummies.
4. "Beginning Git and GitHub : A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer" Mariot Tsitoara , Apress.
5. "Git Prodigy: Mastering Version Control with Git and GitHub" Ebenezer Don.
6. Git Pocket Guide" Richard E. Silverman, 1st Ed, O'Reilly Media.
7. "Learning GitHub" Aidan Feldman, Michaël Sels, and Edward Thomson, 1st Ed, O'Reilly Media.

E-Resources:

1. https://github.com/bobbyiliev/introduction-to-git-and-github-ebook
2. https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
3. https://www.freecodecamp.org/news/git-and-github-for-beginners
4. https://www.datacamp.com/tutorial/github-and-git-tutorial-for-beginners

Assessment Details(both CIE and SEE)

Continuous Internal Assessment of Laboratory/Practical Courses		
Lab Test 1	Lab Test 2	Lab Records
15 marks	15 marks	20 marks
Semester End Examination(SEE)		50 marks

OBJECT ORIENTED MODELLING LAB UISNG STAR UIML

Course Code	**22CDL46B**	**CIE Marks**	**50**
Teaching Hours /Week(L:T:P)	**(0:0:2)**	**SEE Marks**	**50**
Total Hours of Pedagogy	**12 Lab slots**	**Total Marks**	**100**
Credits	**01**	**Exam Hours**	**03**

Course objectives: Basics of object-oriented programming principles and software modeling concepts

This course will enable students to:

- To Write the Problem Statement For the given System.
- To Specify Software Requirement Using Data Flow Diagram.
- To Draw structural and Behavioral diagrams for the given Specifications.

LIST OF LABORATORY PROGRAMS

Perform the following Experiments on following Domains

Identify the software system that need to be Developed.

Document the SRS for the identified System.

Draw the level 0, level 1, and level 2 DFD for the Identified System.

Draw the Class Diagrams and show Various Class Relationships.

Draw the UML Component and Deployment Diagram for the identified System.

Identify the use cases and Develop the Use case Model with include and external Relationships.

Using the identified Scenarios find interaction between objects and represent using Sequence diagram.

Using the identified Scenarios find interaction between objects and represent using Collaboration diagram.

Draw the relevant Activity Diagram for the Same System

Draw the relevant State Chart diagram for the same System

1.	Online Course Reservation System
2.	Railway Reservation System
3.	Exam Registrations
4.	Stock Maintenance Systems
5.	Recruitment Systems
6.	Library Management Systems.
7.	Student Information Systems.
8.	ATM

Course Outcomes:

At the end of the course the student will be able to :

- Design and develop the Problem Statement for the given system
- Analyze the Requirements Specification for an intended Software System Using DFD
- Design and capture the requirements specification for an intended Software system Using use case Modeling
- Design the Structural and behavioral Diagrams for the given specification.
- Design the Sequence Diagram,Activity Diagram and State Chart Diagram for the given specifications.

TEXT BOOKS:

1. **Martin Fowler,** UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison-Wesley Object Technology Series,2003

REFERENCES:

1. Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition,PHI, 2004

E-Resources:

1. StarUML Documentation : https://docs.staruml.io/
2. TutorialsPoint UML Tutorial: https://www.tutorialspoint.com/uml/index.htm
3.GeeksforGeeks UML Tutorial : https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
4.GitHub Repository: https://github.com/staruml

Assessment Details(both CIE and SEE)

Continuous Internal Assessment of Laboratory/Practical Courses		
Lab Test 1	Lab Test 2	Lab Records
15 marks	15 marks	20 marks
Semester End Examination(SEE)		50 marks

BIOLOGY FOR ENGINEERS

Course Code	**22BET47**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P: S)	**3:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	**40**	**Total Marks**	**100**
Credits	**03**	**Exam Hours**	**03**

Pre-requisite: Basic Biology Concepts, Chemistry, Biochemistry

Course objectives:

- To gain the knowledge of Biomolecules.
- Analyze the application of human organ(Devices) system and its bio medical applications
- Study the Human organ systems, develop the bio device and provide engineering solutions.
- To understand the mechanism of bio inspired materials.
- Studying the techniques of Bio-Engineering.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Explanation via real life problem, situation modelling, and deliberation of solutions, hands- on sessions, reflective and questioning /inquiry-based teaching.
2. Instructions with interactions in classroom lectures (physical/hybrid).
3. Use of ICT tools, including YouTube videos, related MOOCs, AR/VR/MR tools.
4. Flipped classroom sessions (~10% of the classes).
5. Industrial visits, Guests talks and competitions for learning beyond the syllabus.
6. Students' participation through audio-video based content creation for the syllabus (asassignments).
7. Use of Gamification tools (in both physical/hybrid classes) for creative learning outcomes.
8. Students' seminars (in solo or group) /oral presentations.

Module – I

BIOMOLECULES AND THEIR APPLICATIONS:

Carbohydrates (cellulose-based water filters, PHA and PLA as bio plastics), Nucleic acids (DNA Vaccine for Rabies and RNA vaccines for Covid19, Forensics – DNA fingerprinting), Proteins (Proteins as food – whey protein and meat analogs, Plant based proteins), lipids (biodiesel, cleaning agents/detergents).

08 Hours

Module – II

HUMAN ORGAN SYSTEMS AND BIO DESIGNS 1 :

Brain as a CPU system (CNS and Peripheral Nervous System, EEG, Robotic arms for prosthetics, Engineering solutions for Parkinson's disease),Eye as a Camera system(bionic eye), Heart as a pump system (Electrical Signaling, ECG, Design of stents, Pace makers, Defibrill ators.

08 Hours

Module – III

HUMAN ORGAN SYSTEMS AND BIO DESIGNS 2:

Lungs as purification system (spirometry, abnormal lung physiology- COPD, Ventilators, Heart-lung machine), Kidney as a filtration system (dialysis System and fabrication), Muscular and Skeletal Systems as scaffolds (Bioengineering solutions for muscular dystrophy and osteoporosis). **08 Hours**

Module – IV

NATURE-BIOINSPIRED MATERIALS AND MECHANISMS :

Echolocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf), Lotus leaf effect (Super hydrophobic and self-cleaning surfaces), Plant burrs (Velcro), Shark skin (Friction reducing swim suits), Kingfisher beak (Bullet train), Human Blood substitutes - hemoglobin-based oxygen carriers (HBOCs) and perflourocarbons (PFCs).

08 Hours

Module – V

TRENDS IN BIOENGINEERING:

Bioprinting techniques and materials, 3D printing of ear, bone and skin. 3D printed foods, Electrical tongue and electrical nose in food Industry, DNA origami and Biocomputing, Bio imaging and Artificial Intelligence for disease diagnosis. Bioremediation and Bio mining via microbial surface adsorption (removal of heavy metals like Lead, Cadmium, Mercury, Arsenic).

08 Hours

Course outcome (Course Skill Set)

At the end of the course, the student will be able to :

1. Apply the concepts of biomolecules to different bioengineering applications.
2. Analyze the functions and mechanism of human organ system to develop the biomimicking devices.
3. Design and implement the bio-engineering Devices to various biomedical techniques.
4. Evaluate and simulate the Bioinspired materials for industrial applications.
5. Build the basic biological concepts via fabricating biodevices using different bio-inspired materials

Assessment Details (both CIE and SEE) Theory Component

Evaluation Type	Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

Suggested Learning Resources:

Books

1. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.
2. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
3. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
4. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
5. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
6. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
7. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
8. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
9. Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N GeethaA C Udayashankar Lambert Academic Publishing, 2019.
10. 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
11. Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016

Web links and Video Lectures (e-Resources):

- https://nptel.ac.in/courses/121106008

- https://freevideolectures.com/course/4877/nptel-biology-engineers-other-non-biologists
- https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009
- https://ocw.mit.edu/courses/20-010j-introduction-to-bioengineering-be-010j-spring-2006
- https://www.coursera.org/courses?query=biology
- https://onlinecourses.nptel.ac.in/noc19_ge31/preview
- https://www.classcentral.com/subject/biology
- https://www.futurelearn.com/courses/biology-basic-concepts

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Group Discussion of Case studies
- Product development, seminar and poster presentations
- Design of novel device/equipment like Cellulose-based water filters, Filtration system

UNIVERSAL HUMAN VALUES

Course Code	**22UHV48**	**CIE Marks**	**50**
Teaching Hours/Week (L:T:P:S)	**1:0:0:0**	**SEE Marks**	**50**
Total Hours of Pedagogy	15 hour Theory Session +15 hour Self study	**Total Marks**	**100**
Credits	**01**	**Exam Hours**	**01**

Prerequisites:

Society, Constitution, Ethics, Environment.

Course objectives:

This introductory course input is intended:

1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

This course is intended to provide a much-needed orientational input in value education to the young enquiring minds.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes.

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematicand rational study of the human being vis-à-vis the rest of existence.
2. In addition to the traditional lecture method, different types of innovative teaching methods may beadopted so that the activities will develop students' theoretical and applied skills.
3. State the need for UHV activities and its present relevance in the society and Provide real-life examples.
4. Support and guide the students for self-study activities.
5. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress in real activities in the field.
6. This process of self-exploration takes the form of a dialogue between the teacher and the students to beginwith, and then to continue within the student in every activity, leading to continuous self evolution.
7. Encourage the students for group work to improve their creative and analytical skills.

Module – I

Introduction to Value Education

Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfill the Basic Human Aspirations. **03 Hours**

Module – II

Harmony in the Human Being

Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health.

03 Hours

Module - III

Harmony in the Family and Society

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.

03 Hours

Module – IV

Harmony in the Nature/Existence

Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence. **03 Hours**

Module – V

Implications of the Holistic Understanding – a Look at Professional Ethics

Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics Holistic Technologies, Production Systems and Management Models- Typical Case Studies, Strategies for Transition towards Value-based Life and Profession

03 Hours

Teaching-Learning Process	Introduction to the concepts- Chalk and talk method, Discussion, Sharing of experiences, Live Examples and videos

Course outcomes:

CO1: Apply a holistic vision of life
CO2: Analyze the socially responsible behavior
CO3: Demonstrate the environmentally responsible work
CO4: Apply ethical human conduct
CO5: Create the Competence and Capabilities for Maintaining Health and Hygiene
Appreciation and aspiration for excellence (merit) and gratitude for all

Assessment Details (both CIE and SEE)

Evaluation Type	Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled down Marks of IAT and CCE to 25
SEE (MCQ Type)		**100**	**50**	**18**	MCQ type question paper of 50 questions. Examination duration in 2 hours
CIE + SEE		**-**	**100**	**40**	

Suggested Learning Resources:

Books

-READINGS:

Text Book and Teachers Manual

a. The Textbook

A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2**nd** Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1

b. The Teacher‟s Manual

Teachers‟ Manual for A Foundation Course in Human Values and Professional Ethics, R R Gaur, RAsthana, G

Reference Books

1. Jeevan Vidya: EkParichaya, A Nagaraj, JeevanVidyaPrakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj – Pandit Sunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)
14. Sussan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986, 1991
15. Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth –Club of Rome‟s report, Universe Books.
16. A Nagraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.
17. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.
18. A N Tripathy, 2003, Human Values, New Age International Publishers.
19. SubhasPalekar, 2000, How to practice Natural Farming, Pracheen (Vaidik) KrishiTantraShodh, Amravati.
20. E G Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers , Oxford University Press
21. M Govindrajran, S Natrajan & V.S. Senthil Kumar, Engineering Ethics (including Human Values), Eastern Economy Edition, Prentice Hall of India Ltd.
22. B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
23. B L Bajpai, 2004, Indian Ethos and Modern Management, New Royal Book Co., Lucknow. Reprinted 2008.

Web links and Video Lectures (e-Resources):

1. Value Education websites, https://www.uhv.org.in/uhv-ii, http://uhv.ac.in, http://www.uptu.ac.in
2. **Story of Stuff, http://www.storyofstuff.com**
3. **Al Gore, An Inconvenient Truth, Paramount Classics, USA**
4. **Charlie Chaplin, Modern Times, United Artists, USA**
5. **IIT Delhi, Modern Technology – the Untold Story**
6. Gandhi A., Right Here Right Now, Cyclewala Productions
7. https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEkQw
8. https://fdp-si.aicte-india.org/8dayUHV_download.php
9. https://www.youtube.com/watch?v=8ovkLRYXIjE
10. https://www.youtube.com/watch?v=OgdNx0X923I
11. https://www.youtube.com/watch?v=nGRcbRpvGoU
12. https://www.youtube.com/watch?v=sDxGXOgYEKM

Activity Based Learning (Suggested Activities in Class)/ Practical learning